

# **Automated Testing of Trello Web Application using Selenium**

## **1. Introduction**

In the ever-evolving world of software development, the importance of effort and reliability cannot be ignored. Modern applications such as the Trello web platform have become indispensable tools for individuals and teams, making management and collaboration easier. Since software applications play an important role in many areas, it is important to ensure their functionality, security, and user experience.

### **Background:**

Trello is a popular project management website used by individuals, teams, and organizations around the world. Its dynamic interface, real-time integration capabilities, and multi-functional organization capabilities make it the first choice for managing both simple and complex tasks. As the user base grows, maintaining the integrity of Trello's work is critical to ensuring consistent and positive user experience.

### **Rationale for Automated Testing:**

Traditional manual testing methods, while valuable, often fall short in handling the intricacies of modern web applications. With the growing complexity of software systems, manual testing can be time-consuming, prone to errors, and challenging to scale efficiently. Automation offers a compelling solution, providing the ability to conduct thorough, repetitive, and rapid testing across various scenarios. For the Trello web application, automation becomes instrumental in ensuring that critical features like user authentication, workspace creation, board management, and list organization function flawlessly.

### **Objective of the Automated Testing Project:**

The purpose of this functional testing is to evaluate the basic functionality of the Trello web application using Selenium WebDriver. We aim to improve the overall security process by automating important test cases including user authentication, office and board creation, and inventory management. By evaluating the system, we aim to detect potential problems, verify the accuracy of user interaction, and identify practices in various applications.

### **Significance of Automation in Testing:**

Automated testing has many advantages such as faster feedback, increased test accuracy and reduced human error. By making recycling data redundant, the development team can focus on innovation and quickly identify ongoing pipeline backlogs. This project uses the power of Selenium WebDriver, a powerful web driver, to create a reliable testing unit that automates the workflow of Trello's core functions.

### **Structure of the Report:**

This report provides information on the process, the application of the content, and the results of our empirical study. It includes identification and prioritization of critical events, establishment of the test environment, development of automatic tests, regression analysis methods and detailed training methods. By exploring these features, this report aims to demonstrate the effectiveness of automation in validating the core functionality of the Trello web application.

In the following sections, we will delve into the experimental method, details of code implementation, backtesting results, introduction to the process, and our conclusion to provide a complete overview of our automated testing process on the Trello platform.

## **2. Test Scenario Identification**

### **2.1 Analysis of Trello Application**

In our analysis of the Trello web application, we conducted a comprehensive examination of its user interface and underlying functionalities. This involved meticulous exploration of the application's various components, including the login page, workspace management features, board creation interface, and list organization mechanisms. Through this analysis, we gained a profound understanding of the application's workflow and identified key areas where the automated testing efforts should be concentrated.

#### **Key Features Explored:**

##### **User Authentication:**

Verification of the user login process, ensuring secure access to the user's account.  
Validation of authentication error handling for incorrect credentials.

##### **Workspace Management:**

Creation of a new workspace, including entering a unique workspace name and an optional description.  
Examination of workspace deletion functionality, ensuring it removes the workspace and its associated boards and lists.

##### **Board Creation and Management:**

Creation of new boards within a workspace, focusing on various board types and templates.  
Verification of board editing capabilities, such as changing the board name and background color.  
Exploration of board deletion functionality, ensuring it removes the board and its associated lists.

##### **List Organization:**

Creation of new lists within a board, validating that new lists can be added.  
Testing the ordering and reordering of lists, ensuring their proper display and arrangement on the board interface.

### **2.2 Test Scenario Identification**

Based on our analysis, we identified the following critical test scenarios to be executed using Selenium WebDriver. These scenarios encompass both positive and negative test cases, allowing us to evaluate the application's behavior under various conditions.

#### **Identified Test Scenarios:**

##### **User Authentication:**

Positive Scenario: Verify successful login with valid credentials.  
Negative Scenario: Validate error message display for incorrect login attempts.

##### **Workspace Management:**

Positive Scenario: Test the creation of a new workspace with a unique name and optional description.  
Negative Scenario: Verify error handling when attempting to create a workspace with an existing name.

### **Board Creation and Management:**

Positive Scenario: Validate the creation of a new board within a workspace.

Positive Scenario: Test editing features, including changing board name and background color.

Negative Scenario: Verify error handling when creating a board without a name.

### **List Organization:**

Positive Scenario: Test the creation of new lists within a board.

Positive Scenario: Validate the proper ordering and reordering of lists on the board interface.

Negative Scenario: Verify error handling when attempting to create a list with a blank name.

These test scenarios were identified based on their significance in ensuring the functionality, usability, and reliability of the Trello web application. Each scenario will be automated and rigorously tested using Selenium WebDriver, allowing us to validate the application's behavior and enhance its overall quality.

## **3. Test Script Creation**

### **3.1 Testing Environment Setup**

Selenium WebDriver is installed by ChromeDriver, using the Java programming language.

Make sure you are waiting to sync with the website.

### **3.2 Automated Test Script Development**

#### **Main Class**

The Main class serves as the entry point for the automation process. Streamlines the testing process by creating mock-up of logins, workstations, dashboards, and checklists. It went to the Trello login page, created a login and created a workspace, a board, and a checklist.

#### **Login Class**

The login class is used to manage user authentication, login email, password, and job login. Use clear instructions to verify the presence of user interface elements.

#### **Workspace Class**

The Workspace class helps your manage the creation of new workspaces. Enters the workspace name and description, simulating the functions of the keyboard actions for efficient work.

#### **Boards Class**

The Boards class help create new boards in the workspace. It went to the Boards section, and started the board creation process, entered the name of board, and submitted the board creation form.

#### **List Class**

The List class controls the creation of new names in the panel. Shows the list of entries, enter new entries, and confirms the entry using the keyboard.

## **4. Regression Testing**

Regression testing is done by re-running the automated test scenarios after changes or update is made to the application. Data inconsistencies or flaws found during evaluations are returned for further evaluation.

## **5. Reporting**

Reporting Functionality: Automated testing uses the reporting function by generating a complete test report showing test results, including pass and failure.

Visualization: Include visual representations such as charts and graphs were incorporated for clear understanding.

## **6. Test Execution and Documentation**

### **6.1 Test Execution Process**

#### **1. Automated execution:**

Automated tests including user identification, office management, Kanban creation and management, and record organization diagnostics are completed to complete Trello Web application operations. Tests run across multiple browsers and platforms to ensure cross-browser compatibility and platform independence. Selenium WebDriver manages interaction with web content by simulating user interaction in the technology.

#### **2. Continuous integration:**

The electronic tester is integrated into the continuous circuit (CI) line to ensure tests are always performed. This integration provides rapid feedback to development teams, allowing them to detect and resolve issues immediately. CI integration also plays an important role in catching regressions and ensuring that new changes do not impact existing functionality.

#### **3. Parallel Test Execution:**

To optimize testing time, test suites are configured to be executed in parallel. Parallel execution allows multiple tests to be run simultaneously in different execution programs, reducing overall execution time. This approach is especially important when working with large laboratories to provide timely feedback without interrupting testing.

### **6.2 Test Files**

#### **1. Test description:**

Each test corresponds to a specific test and contains detailed information. The description of the test case includes the steps to be taken, the expected results, and any special conditions or prerequisites for the test to be successful. Clear and concise explanations help testers and developers understand test scenarios and desired behaviors.

#### **2. Results and Lists:**

Exam results are generated after each test is completed. The results include a summary of the number of successful attempts, the number of successful and unsuccessful tests, and the overall success rate. Detailed records are also kept recording information about each step taken during the test. These logs can be useful for diagnosing faults and understanding the part of the process that caused a particular event.

## **7. Conclusion and Recommendations**

### **7.1 Results of the project**

#### **1. Basic Functionality Verification:**

Testing successfully validated the basic functionality of the Trello web application. By analyzing the systematic situation, the main points of the application, such as the development of powerful scripts and tests, user identification, site management work, board of directors and organization manifesto, are carefully examined. The automation package ensures the integrity of the simple operation of the application by providing fast and reliable feedback.

#### **2. Repeatability testing:**

Integrating integrated circuit breakers into your continuous improvement (CI) line increases e

efficiency. Automated testing is performed each time a submission is made, quickly identifying backlogs and ensuring that existing projects are not affected by code updates. Parallel execution capabilities further optimize testing time, allow for feedback time, and support rapid development.

### **3. Error detection and reporting:**

Automated testing effectively identifies problems and inconsistencies in behavior. Detailed error reports are generated for each identified issue, including a brief description, screenshots, and additional details. This detailed information provides developers with the information they need to quickly diagnose and resolve problems. Bug reports are a useful resource for tracking the status of reported issues and ensuring they are resolved.

### **4. Improve collaboration and communication:**

Transparent communication and regular feedback support effective collaboration between testers, developers, and stakeholders. Stakeholders receive notifications of past testing, test results, and discovered issues, allowing them to make informed decisions. An ongoing feedback loop ensures that the testing process is tailored to the needs of stakeholders, incorporating their insights and needs into the testing process.

In summary, automated testing of Trello web applications using Selenium WebDriver improved the quality assurance process. The project recognized the importance of the work, facilitated measuring the effectiveness of the recovery, and facilitated coordination of stakeholders. Key lessons learned and recommendations highlight the importance of continuous monitoring, error management, research and testing to ensure good usability. Testing efforts will continue in line with the Hills landscape development software, with future instructions on test suite expansion, CI/CD integration, and mobile app testing.